REMARKS

I. General

Claims 1-37 were pending in the present application, and all of the pending claims are rejected in the current Office Action (mailed September 22, 2004). The outstanding issues raised in the current Office Action are:

- Claim 5 is rejected under 35 U.S.C. § 112, second paragraph; and

- Claims 1-37 are rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,775,692 issued to Albert et al (hereinafter "*Albert*").

In response, Applicant respectfully traverses the outstanding claim rejections, and requests reconsideration and withdrawal thereof in light of the amendments and remarks presented herein.

II. Amendments

Claim 5 is amended herein. No new matter is added by this claim amendment. More specifically, claim 5 is amended to delete the phrase "in step e)" to resolve the lack of antecedent basis issue identified in the present Office Action. This is not intended to be a narrowing amendment.

III. Rejection Under 35 U.S.C. § 112, Second Paragraph

Claim 5 is rejected under 35 U.S.C. § 112, second paragraph. Claim 5 is rejected because there is insufficient antecedent basis for the "step e)" recited therein. As described above, claim 5 is amended herein in a manner that corrects this deficiency. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 5 under 35 U.S.C. § 112, second paragraph.

IV. Rejections Under 35 U.S.C. § 102(e)

Claims 1-37 are rejected under 35 U.S.C. § 102(e) as being anticipated by *Albert*. Applicant respectfully traverses this rejection as provided further below.

To anticipate a claim under 35 U.S.C. § 102, a single reference must teach every element of the claim, *see* M.P.E.P. § 2131. Applicant respectfully submits that *Albert* fails to teach each and every element of claims 1-37.

Independent Claim 1

Albert fails to teach all elements of independent claim 1. As described further below, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 1. Accordingly, without conceding that *Albert* teaches any of the other elements of claim 1, *Albert* fails to teach at least those elements described further below.

For example, independent claim 1 recites "b) handing off said communication session to a selected server computer from said first server computer over a persistent control channel using TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer, that implement a TCP handoff protocol that works within kernel levels of an existing TCP/IP protocol" (emphasis added). *Albert* fails to teach TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems, as recited in claim 1.

The current Office Action cites col. 14, line 65 – col. 15, line 27 (SYN/ACK packets) of *Albert* in support of its assertion that *Albert* teaches this element of claim 1, *see* page 3 of the Office Action. Col. 14, line 65 – col. 15, line 27 of *Albert* provides:

> FIG. 5 is a diagram illustrating how a service manager provides instructions to two separate forwarding agents for handling a connection. A client 500 sends a SYN packet to a first forwarding agent 502. Forwarding agent 502 has previously received a wildcard affinity from a service manager 504 on a dedicated connection on which service manager 504 multicasts wildcard affinities to forwarding agents. As a result of the wildcard match, forwarding agent 502 encapsulates the SYN packet and forwards it to service manager 504. Service manager 504 receives the SYN packet and returns it to forwarding agent 502 along with a fixed affinity specifying an action to be performed on the packet. The action defined in this example is translating the destination IP address of the packet from a virtual IP address to the IP address of a host 506. Hosts 506 and 507 together implement a virtual machine 510.
>
> Host 1 receives the SYN packet from forwarding agent 1 and returns a SYN ACK packet back to client 500. However, for some reason, the SYN ACK packet from host 1 is routed not through forwarding agent 502, but instead through forwarding agent 512. Forwarding agent 512 receives the SYN ACK

> and notes that it matches a wildcard affinity corresponding to the flow of packets from host 506 to client 500. Forwarding agent 512 encapsulates the SYN ACK packet and sends it to service manager 504. Service manager 504 defines an action for the SYN ACK packet and includes that action in a second fixed affinity which it sends along with the encapsulated SYN ACK packet back to forwarding agent 512. Forwarding agent 512 then sends the SYN ACK packet on to client 500 where it is processed.

The above portion of *Albert* teaches that upon receiving a SYN packet, the forwarding agent 502 forwards such SYN packet to a service manager if a predefined wildcard affinity matches the SYN packet. The service manager returns the SYN packet to the forwarding agent with a fixed affinity specifying an action to be performed on the packet. Host 1 receives the SYN packet from the forwarding agent 502 and returns a SYN ACK packet through forwarding agent 512 to client 500. Again, because the SYN ACK packet matches a predefined wildcard affinity, forwarding agent 512 sends the SYN ACK packet to the service manager. The service manager returns the SYN ACK packet to the forwarding agent 512 with a fixed affinity, and the forwarding agent 512 sends the SYN ACK packet on to client 500.

The above portion of *Albert* fails to teach at least the above identified element of claim 1. That is, the above portion of *Albert* in no way teaches "TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer". For instance, assuming that the Office Action contends the forwarding agent 502 of *Albert* to be the recited first server computer and the Host 1 of *Albert* to be the recited selected server computer, *Albert* fails to teach that either of such elements of *Albert* use a TCP handoff module that is dynamically loadable within a TCP/IP stack in its operating system.

Accordingly, *Albert* fails to teach at least the above-identified element of claim 1, and therefore claim 1 is not anticipated under 35 U.S.C. § 102 by *Albert*.

Independent Claim 11

Albert fails to teach all elements of independent claim 11. As described further below, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 11. Accordingly, without conceding that *Albert* teaches any of the

other elements of claim 11, *Albert* fails to teach at least those elements described further below.

First, independent claim 11 recites "monitoring traffic associated with establishing said TCP/IP communication session to understand a first initial TCP state of said first server computer associated with said TCP/IP communication session, at a first bottom-TCP (BTCP) module at said first server computer" (emphasis added). *Albert* fails to teach a first bottom TCP (BTCP) module at the first server computer. Independent claim 11 further recites "a second BTCP module at said selected server computer". *Albert* fails to teach a second bottom TCP (BTCP) module at the selected server computer. Indeed, *Albert* does not teach a modularized solution, and thus fails to teach any modules whatsoever.

Additionally, independent claim 11 further recites "e) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content" (emphasis added). *Albert* fails to teach determining a web server that can best process a received web request based on the content of such web request. Rather, as described further below, *Albert* teaches using pre-defined wildcard affinities to select a server to handle a request based on information included in a SYN packet, such as the requesting client's IP address, rather than selecting a server based on the content of a web request.

The Office Action cites col. 9, lines 10-34 and 45-58 of *Albert* in support of its assertion that *Albert* teaches this element of claim 11, *see* page 11 of the Office Action. Col. 9, lines 10-34 and 45-58 of *Albert* provides:

> In addition to specifying instructions for each flow, service managers must also obtain information about each new flow from the forwarding agents. For example, when a service manager provides load balancing through a set of forwarding agents, the service manager uses fixed affinities to provide specific instructions to the forwarding agents detailing where packets for each load balanced flow are to be forwarded. In addition to providing those specific instructions, the service manager also provides general instructions to each forwarding agent that specify which new flows the service manager is interested in seeing. These general instructions are provided using wildcard affinities. Wildcard affinities, which are described in detail below, specify sets of flows that are of interest to a service manager. In one embodiment, this is done by specifying subnet masks that determine sets of source and destination IP addresses that will be forwarded to a service manager. In addition, ports or sets of ports and protocol may be specified in wildcard affinity as well. As is described further below, the use of wildcard affinities enables separate service

> managers to be configured to provide services for different sets of flows. Each service manager specifies the flows of interest to it and other service managers handle other flows. In this manner, service managers can be configured in parallel to share load.
>
> * * *
>
> In the case of load balancing, service managers send wildcard affinities to forwarding agents. The wildcard affinities specify destination IP addresses that correspond to virtual IP addresses of server clusters that are to be load balanced by the service manager. The forwarding agents then forward new packets sent to those virtual IP addresses to the appropriate service manager. The service manager selects a server from the server cluster and then the service manager sends a fixed affinity to each forwarding agent that instructs the forwarding agent to forward packets for that specific flow to the selected server in the cluster. Forwarding agents may also forward packets for purposes other than load balancing. Packets may be forwarded to real IP addresses as well as virtual IP addresses.

Albert does not teach determining a web server that can best process a received web request based on the content of such web request. Rather, *Albert* teaches pre-setting a wildcard affinity based on a client's IP address. For instance, the above portion of *Albert* teaches that wildcard affinities specify sets of flows that are of interest to a service manager. The wildcard affinities are pre-selected (e.g., before even receiving a request from a client), to specify a subnet mask that identifies a set of source and destination IP addresses. Thus, for instance, a particular IP address corresponding to a client of interest can be identified by a wildcard affinity.

As described further in *Albert* at col. 12, line 6 – col. 14, line 15, a Syn packet is used to identify whether the flow matches a wildcard affinity, in which case it is forwarded to the service manager for determination of how to handle the flow. Thus, the service manager in *Albert* selects a server responsive to receipt of a Syn packet, which is an initial connection establishment packet sent before it is even known what the request will be (i.e., before knowing that it is a web request). That is, the Syn packet upon which *Albert* selects a server, does not include content of a web request. Rather, the Syn packet includes source and destination IP addressed, from which it is determined by the forwarding agents whether such packet corresponds to a pre-set wildcard affinity. Thus, the back-end server is not selected in *Albert* based on the content of such web request, as the server is selected based on other information (e.g., source IP address) included in the Syn packet.

Further, claim 11 recites "k) sending response packets from said selected server computer directly to said client computer by changing said response packets to reflect said first TCP state and a first IP address of said first server computer" (emphasis added). *Albert* fails to teach a system in which the selected server responds directly to the client computer, but rather all communication flows through the forwarding agents. That is, the forwarding agents and service manager of *Albert* act as a front-end node through which all communication from clients is received in order to determine the back-end server (or host) to which the communication should be sent, and all return communication from the back-end server (or host) is sent back through the forwarding agents to the client. Thus, this configuration is similar to that of FIGURE 1 described in the present application, wherein the forwarding agents/service manager of *Albert* provide a load balancer through which all communication between the clients and the back-end servers (hosts) flows. *Albert* does not teach a configuration in which a selected back-end server sends a response directly to the client computer.

Accordingly, *Albert* fails to teach at least the above-identified elements of claim 11, and therefore claim 11 is not anticipated under 35 U.S.C. § 102 by *Albert*.

Independent Claim 26

Albert fails to teach all elements of independent claim 26. As described above, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 26. Accordingly, without conceding that *Albert* teaches any of the other elements of claim 26, *Albert* fails to teach at least those elements described further below.

For example, independent claim 26 recites "an upper TCP (UTCP) module located above a TCP module in an operating system of said server computer" (emphasis added). *Albert* fails to teach such an upper TCP (UTCP) module located above a TCP module in an operating system of a server computer.

Further, claim 26 recites "a bottom TCP (BTCP) module located below said TCP module" (emphasis added). *Albert* also fails to teach such a bottom TCP (BTCP) module located below the TCP module in an operating system.

Indeed, *Albert* does not teach modules at all in the operating systems of its computers. While *Albert* describes forwarding agents, service manager, and host computers, it fails to teach that any of those include a module located above or below a TCP module in an operating system.

Accordingly, *Albert* fails to teach at least the above-identified element of claim 26, and therefore claim 1 is not anticipated under 35 U.S.C. § 102 by *Albert*.

Dependent Claims

In view of the above, Applicant respectfully submits that independent claims 1, 11, and 26 are not anticipated under 35 U.S.C. § 102 over *Albert*. Further, each of dependent claims 2-10, 12-25, and 27-37 depend either directly or indirectly from one of independent claims 1, 11, and 26, and thus inherit all limitations of the respective independent claim from which they depend. It is respectfully submitted that dependent claims 2-10, 12-25, and 27-37 are allowable not only because of their dependency from their respective independent claims for the reasons discussed above, but also in view of their novel claim features (which both narrow the scope of the particular claims and compel a broader interpretation of the respective base claim from which they depend).

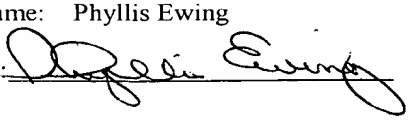
V. Conclusion

In view of the above, Applicant believes the pending application is in condition for allowance. Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 08-2025, under Order No. 10010812-1 from which the undersigned is authorized to draw.

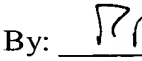
I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV 482745208US in an envelope addressed to: M/S Amendment, Commissioner for Patents, Alexandria, VA 22313.

Date of Deposit: December 22, 2004

Typed Name: Phyllis Ewing

Signature: 

Respectfully submitted,

By: 

R. Ross Viguet
Attorney/Agent for Applicant(s)
Reg. No. 42,203
Date: December 22, 2004
Telephone No. (214) 855-8000